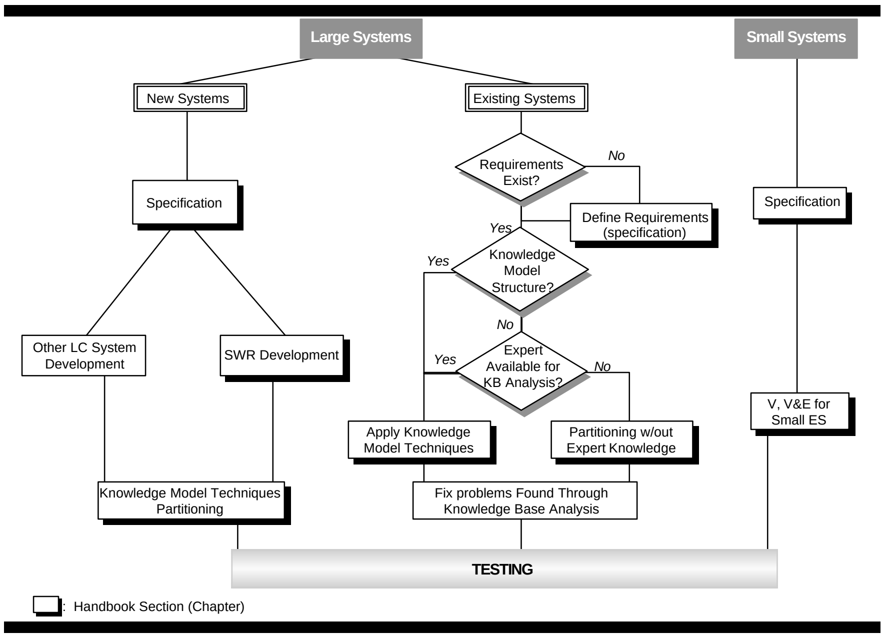
Verification, Validation, and Evaluation of Expert Systems

Volume I

An FHWA Handbook

1st Edition (Ver. 1.1)



By:

James A Wentworth
Chief; Advance Research Team
FHWA, McLean VA

Rodger Knaus
Senior Research Scientist
MiTech, Rockville MD

Hamid Aougab
Senior Research Engineer
RAM, McLean VA

August 1996

1. Report No. FHWA-RD-95-196	2. Government Accession No.	3. Recipient's Catalog No
4. Title and Subtitle VERIFICATION, VALIDATION AND EVALUATION OF EXPERT SYSTEMS Volume 1		5. Report Date 6. Performing Organization Code
7. Author(s) James A. Wentworth, Rodger Knaus, Hamid Aougab		8. Performing Organization Report No.
9. Performing Organization Name and Address MiTech Incorporated 9430 Key West Avenue, Suite 100 Rockville, Maryland 20860 Federal Highway Administration Washington, DC		10. Work Unit No. (TRAIS) 11. Contract or Grant No. DTFH61-93-Z-00043
12. Sponsoring Agency Name and Address Federal Highway Administration Office of Safety and Traffic Operations R&D 6300 Georgetown Pike McLean, Virginia 22101-2296		13. Type of Report and Period Covered Interim Report April 1993-August 1995 14. Sponsoring Agency Code

15. Supplementary Notes

Contracting Officer's Technical Representative (COTR): James A. Wentworth, HSR-1

This research was funded by the Federal Highway Administration

16. Abstract

The importance and difficulty of performing verification, validation and evaluation (VV&E) on expert systems cannot be overstated. This is one of the major factors slowing the development and acceptance of expert systems in the transportation community. There is little agreement among experts on how to accomplish the VV&E of expert systems. The complexity and uncertainty related to these tasks has lead to the situation where most expert systems are not adequately tested. In some cases testing is ignored until late in the development cycle, with predictable negative results.

This guide discusses how VV&E should be incorporated into the expert system lifecycle, shows how to partition knowledge bases with and without expert domain knowledge, presents knowledge models, presents methods for validating the underlying experts' knowledge, and presents management issues related to expert systems development and testing. Mathematical proofs for partitioning, consistency, and completeness and visualization of concepts are presented.

17. Key Words	18. Distribution Statement
verification, validation, evaluation, expert systems, knowledge base, knowledge engineering, partitioning, consistence, completeness.	No restriction. This document is available to the public through the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (Of this report)	20. Security Classifi. (Of this page)	21. No. Of Pages	22. Price
Unclassified	Unclassified		

Dedication

In memory of Roger E. Hoffman, our deceased colleague whose decency, honor, humor and intellect touched all who worked with him. The Hoffman Regions described in this handbook, a crucial concept in defining the completeness and consistency of expert systems, were conceived by Roger.

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Figures and Tables

LIST OF FIGURES

Figure

LIST OF TABLES

Table

1. Introduction

Roadway engineering and construction pre-date Roman times. Over the centuries, standards in design and construction and the documentation of practice have been raised to very high levels. In the process of modernizing and improving design, construction, and maintenance, new approaches and technologies have been incorporated into civil engineering practice. Initially, many of the new technologies did not achieve the levels of reliability and standardization required by the civil engineering profession. Regrettably, many expert systems fall into this category, due partly to the lack of verification, validation, and evaluation standards.

The goals of expert systems are usually more ambitious than those of conventional or algorithmic programs. They frequently perform not only as problem solvers but also as intelligent assistants and training aids. Expert systems have great potential for capturing the knowledge and experience of current senior professionals and making the expert's wisdom available to others in the form of training aids or technical support tools. Applications include design, operations, inspection, maintenance, training, and many others.

In traditional software engineering, testing [verification, validation and evaluation (VV&E)] is claimed to be an integral part of the design and development process. However, in the field of expert systems, there is little consensus on what testing is necessary or how to perform it. Further, many of the procedures that have been developed are so poorly documented that it is difficult, if not impossible, for them to be reproduced by anyone other than the originator. Also, many procedures used for VV&E were designed to be specific to the particular domain in which they were introduced. The complexity and uncertainty related to these tasks has led to a situation where most expert systems are not adequately tested.

Impelled by the existing environment of lack of consensus among experts and inadequate procedures and tools, the FHWA developed this guideline for expert system verification, validation, and evaluation, complete with software to implement recommended techniques. The guideline is needed because knowledge engineers today do not often design and carry out rigorous test plans for expert systems. The software is necessary because real-world knowledge bases containing hundreds of rules and dozens of variables are difficult for humans to assimilate and evaluate. Computerized verification and validation (V&V) tools would also enable the knowledge engineer to use interim V&V reports to guide knowledge acquisition and coding, something that is too labor-intensive with hand methods. The techniques presented represent a workable solution to a difficult problem. Hopefully they also provide a basis for further enhancements and improvements.

Basic Definitions

This guide covers *verification, validation, and evaluation* of *expert systems.* An expert system is a computer program that includes a representation of the experience, knowledge, and reasoning processes of an expert. Figure 5.1 shows a six rule expert system that will be used as an example throughout this guide.

Introduction

Verification of an expert system, or any computer system for that matter, is the task of determining that the system is built according to its specifications. *Validation* is the process of determining that the system actually fulfills the purpose for which it was intended. *Evaluation* reflects the acceptance of the system by the end users and its performance in the field. In other words *(Miskell et al, 1989)*:

- **Verify** to show **the system is built right.**
- **Validate** to show **the right system was built.**
- **Evaluate** to show **the usefulness of the system**.

Verification

As stated above, verification asks the question "is the system built right?," i.e., verification is checking that the knowledge base is complete and that the inference engine can properly manipulate this information. Issues raised during verification include:

- Does the design reflect the requirements? Are all of the issues contained in the requirements addressed in the design?
- Does the detailed design reflect the design goals?
- Does the code accurately reflect the detailed design?
- Is the code correct with respect to the language syntax?

When the program has been verified, it is assured that there are no "bugs" or technical errors.

Validation

Validation answers the question "is it the right system?" "is the knowledge base correct?" or "is the program doing the job it was intended to do?" Thus, validation is the determination that the completed expert system performs the functions in the requirements specification and is usable for the intended purposes. It is impossible to have an absolute guarantee that a program satisfies its specification, only a degree of confidence that a program is valid can be obtained. Issues addressed during validation of an expert system include:

- How well do inferences made compare with knowledge and heuristics of experts in the field?
- How well do inferences made compare with historic (known) data?
- What fraction of pertinent empirical observations can be simulated by the system?
- What fraction of model predictions are empirically correct?
- What fraction of the system parameters does the model attempt to mimic?

Evaluation

Evaluation addresses the issue "is the system valuable?" This is reflected by the acceptance of the system by its end users and the performance of the system in its application. Pertinent issues in evaluation are:

- Is the system user friendly, and do the users accept the system?
- Does the expert system offer an improvement over the practices it is intended to supplement?
- Is the system useful as a training tool?
- Is the system maintainable by other than the developers?

To illustrate the difference, the task might be to build a system that computes the serviceability coefficient of pavement. The specifications for the system are contained in textbooks that define the coefficient. To validate the system one must test the serviceability of the program on examples in the texts and other test cases and compare the results of the program with independently computed coefficients for the same examples. It is important to use a test set that covers all the important cases and contains enough examples to make sure that correct results are not just anomalies.

Once the system is validated, the next step is to verify it. This involves completeness and consistency checks and examining for technical correctness using techniques such as are described in this handbook. The final step is evaluation. For the serviceability program, this means giving the system to engineers to use in computing the coefficient. Although the system is known to produce the correct result, it could fail the evaluation because it is too cumbersome to use, requires data that are not readily available, does not really save any effort, does something that can be estimated accurately enough without a computer, solves a problem rarely needed in practice, or produces a result not universally accepted because different people define the coefficient in different ways.

Need for V&V

It is very important to verify and validate expert systems as well as all other software. When software is part of a machine or structure that can cause death or serious injury, V&V is especially critical. In fact, there have already been failures of expert systems and other software that have resulted in death. For example, a robotized overhead material mover struck an overhead crane at an Alcoa aluminum plant, killing the crane operator, because its narrow-field vision system saw only an interior region of the crane front, a blank field to the robot. In another case, a much-patched system for cancer radiation treatment gave a fatal dose to at least one patient, because the operator overrode the emergency stop; it had given repeated false alarms in past situations.

Introduction

Expert systems use computational techniques that involve making guesses, just as human experts do. Like human experts, the expert system will be wrong some of the time, even if the expert system contains no errors. The knowledge on which the expert system is based, even if it is the best available, does not completely predict what will happen. For this reason alone, it is important for the human expert to validate that the advice being given by the expert system is sound. This is especially critical when the expert system will be used by persons with less expertise than the expert, who can not themselves judge the accuracy of the advice from the expert system.

In addition to mistakes which an expert system will make because the available knowledge is not sufficient for prediction in every case, expert systems contain only a limited amount of knowledge concentrated in carefully defined knowledge areas. Today's expert systems have no common sense knowledge. They only "know" exactly what has been put into their knowledge bases. There is no underlying truth or fact structure to which it can turn in cases of ambiguity. This means that an expert system containing some errors in its knowledge base can make mistakes that would seem ridiculous to a human, and not realize that a mistake had occurred. [On the other hand, expert systems do not get tired or sick or bored or fall in love, and therefore avoid some of the "careless" mistakes that a person might make, particularly on repetitive problems.] If the expert system does not realize its mistake, and it is being used by a person with limited expertise, there is nobody to detect the error. Therefore, where the expert system is going to be used by someone without expertise, and the decisions made have the potential for harm if made badly, the very best effort at verification and validation is required.

Problems in Implementing Verification, Validation, and Evaluation for Expert Systems

One of the impediments to a successful V&V effort for expert systems is the nature of expert systems themselves. Expert systems are often employed for working with incomplete or uncertain information or "ill structured" situations (Giarratano and Riley, 1989). These are cases where, as in a diagnostic expert system, not all symptoms for all malfunctions are known in advance. In these situations, reasoning offers the only hope for a good solution. Since expert system specifications often do not provide a precise criterion against which to test, there is a problem in verifying, validating, and evaluating expert systems according to the definitions in section 1. For example, specifying that a speech recognition system should understand speech does not define a testable standard for the system. Some vagueness in the specifications for expert systems is unavoidable; if there are precise enough specifications for a system, it may be more effective to design the system using conventional programming languages.

Another problem in VV&E for expert systems is that expert system languages are not structured to accommodate the relatively unstructured applications. However, rigid structure in implementing the code is a key technique used in writing verifiable code, such as the Cleanroom approach.

Cleanroom software specification (Linger, 1993) begins with a specification of required system behavior and architecture. Many expert systems cannot conform to the rigidity required by this quality control method used principally for conventional programming.

Introduction

Intended Audiences for the Handbook

The following table describes the intended audiences for the handbook, and the parts of the handbook that will be most useful to these audiences:

Table 1-1: Intended Audiences for the Handbook

Audience	**Task to be Performed**	**Part of Handbook**
Managers	Manage expert system project	Chp. 1: Introduction Chp. 3: Planning And Management
Knowledge Engineers	Build new expert systems	Chp. 4: Developing a Verifiable System Chp. 7: Knowledge Modeling Chp. 9: Validating Undelying Knowledge
Knowledge Engineers	Perform VV&E on existing systems	Chp. 5: The Basic Proof Method Chp. 6: Finding Partitions without Expert Knowledge Chp. 8: VV&E for Small Systems Chp. 9: Validating Undelying Knowledge
Highway Engineers	Ensure that a correct new expert system is built	Chp. 3: VV&E on New Systems Chp. 10: Testing Chp. 11: Evaluation & Manag. Issues
Highway Engineers	Ensure that an existing expert system has been validated	Chp. 3: VV&E on Existing Systems Chp. 10: Testing Chp. 11: Evaluation & Manag. Issues
Software Researchers	Critique and extend VV&E methods	Chp. 2: V&V: Past Practice Chp. 6: Finding Partitions without Expert Knowledge Chp. 7: Knowledge Modeling Chp. 8: VV&E for Small Systems Chp. 9: Validating Undelying Knowledge

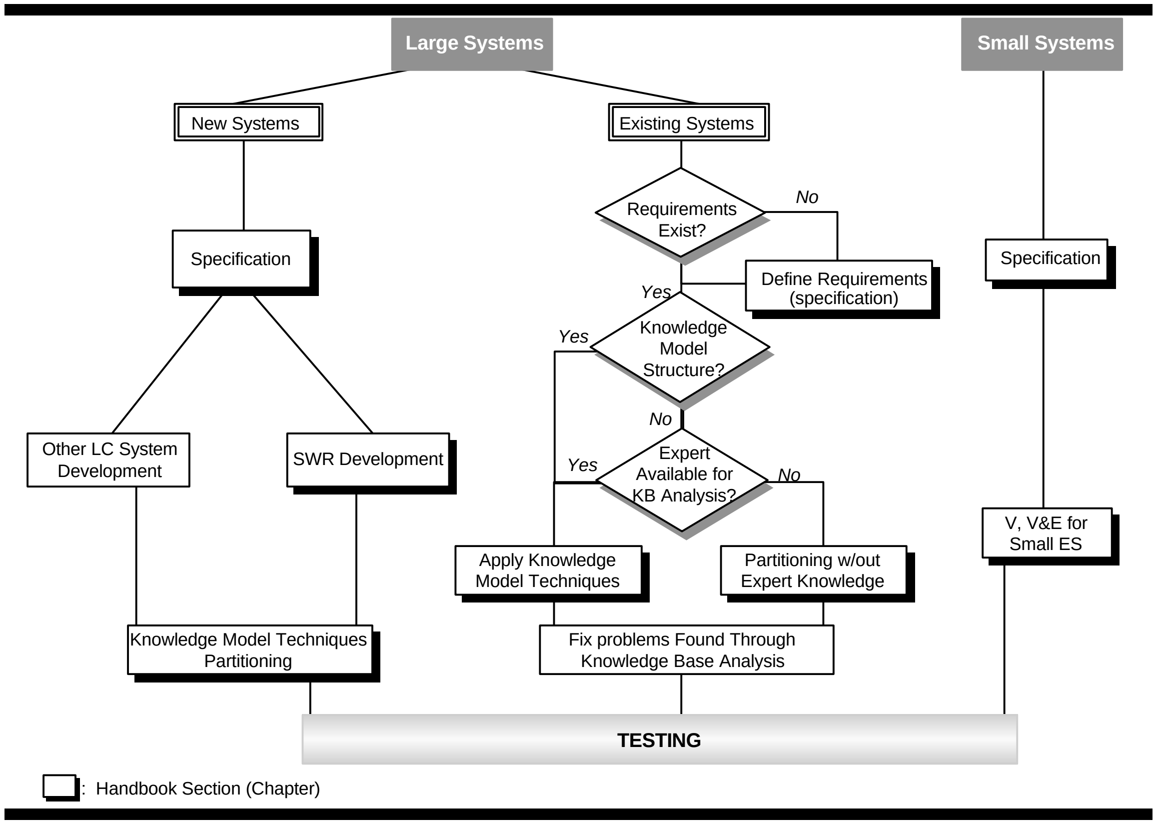


Figure 1.1: The V&V Process